Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2004-08

SOLVING A DYNAMIC AND STOCHASTIC VEHICLE ROUTING PROBLEM WITH A SAMPLE SCENARIO HEDGING HEURISTIC

Lars M. Hvattum
Arne Løkketangen
Gilbert Laporte

April 2004

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone : (514) 343-7575 / Télécopieur : (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

104-04344/

PUBLICATION
CRT-2004-08

SOLVING A DYNAMIC AND STOCHASTIC VEHICLE ROUTING PROBLEM WITH A SAMPLE SCENARIO HEDGING HEURISTIC

by

Lars M. Hvattum[1]
Arne Løkketangen[1]
Gilbert Laporte[2]

April 2004

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

[1] Molde University College, Britveien 2, Molde, 6411, Norway (Lars.M.Hvattum, Arne.Lokketangen@himolde.no)

[2] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7, and Canada Research Chair in Distribution Management, HEC Montréal, 3000, Côte-Sainte-Catherine, Montréal, Canada H3T 2A7 (gilbert@crt.umontreal.ca)

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2004

Abstract

The statement of the standard *Vehicle Routing Problem* (VRP) cannot always capture all aspects of real-world applications. As a result, extensions or modifications to the model are warranted. Here we consider the case when customers can call in orders during the daily operations, i.e., both customer locations and demands may be unknown in advance. This is modelled as a combined dynamic and stochastic programming problem, and a heuristic solution method is developed where sample scenarios are generated, solved heuristically, and combined iteratively to form a solution to the overall problem.

Key words: vehicle routing problem, time windows, dynamic, stochastic, scenarios.

Résumé

L'énoncé du problème standard de tournées de véhicules (PTV) ne peut pas tenir compte de toutes les complications qui se présentent en pratique. En conséquence, on doit résoudre des extensions ou des modifications de ce problème. Dans cet article, on considère le cas où les clients placent des appels pendant la journée; les localisations des clients et leur demande sont inconnues à l'avance. On modèle ce cas comme un problème combiné de programmation dynamique et stochastique. On développe une heuristique dans laquelle des scénarios sont générés aléatoirement, résolus de façon approximative et combinés de façon itérative pour produire une solution au problème global.

Mots clés : problème de tournées de véhicules, fenêtres de temps, dynamique, stochastique, scénarios.

1 Introduction

The standard *Vehicle Routing Problem* (VRP) is central to distribution management, and much effort has been expended to develop efficient exact algorithms and heuristics for it (see, e.g., Toth and Vigo, 2002). In real-world settings the VRP statement usually fails to capture all essential aspects of the problem at hand. Hence, a large number of extensions have been studied, e.g., adding time windows, different kinds of customers, split deliveries, heterogeneous vehicle fleets, and more, which all yield richer and more useful formulations.

One aspect of real-world problems that necessitates modifications is uncertainty, i.e., some of the parameters of the model are initially unknown, or known only probabilistically. Two modelling approaches put forward to handle such conditions are *Dynamic Vehicle Routing Problems* (DVRPs) and *Stochastic Vehicle Routing Problems* (SVRPs), respectively. Due to recent technological advances in communications and information technologies these problem classes become increasingly important, and the potential savings generated by adapting routing decisions to dynamic or stochastic contexts are substantial (see Psaraftis, 1995, and Wahl et al., 2002).

In this paper we consider, based on an actual case from a large distribution company, a problem that can be seen as a mix between a DVRP and an SVRP. Customers can place orders at any time during the day, i.e., the set of customers is at first partially unknown. Vehicle routes servicing the customers can therefore be expected to change during the day to accommodate new orders. From historical data, probability distributions for the attributes of the unknown customers may be generated and thereafter used during the planning of the vehicle routes. We show how such a problem can be formulated as a multi-stage stochastic programming problem with recourse. We develop an efficient multi-stage heuristic that gradually constructs routes by exploiting statistical information on future customer demand. This heuristic is shown to yield significant savings with respect to a more standard pure dynamic heuristic.

The remainder of this paper is organized as follows. The concepts of dynamic and stochastic vehicle routing will be examined in Section 2. In Section 3 the real-world problem is introduced. Section 4 contains a mathematical formulation of the problem. A sample scenario hedging heuristic is presented in Section 5. Analyses and computational results follow in Section 6, before conclusions and suggestions for future work are outlined in Section 7.

2 Dynamic and Stochastic Vehicle Routing

Before discussing DVRPs and SVRPs we will briefly describe the classical capacitated static VRP: a set of customers requiring deliveries is served by a fleet of identical vehicles. The goal is to create vehicle tours of minimum total length such that each vehicle starts and ends at the depot, every customer is visited exactly once, and the sum of demands of the customers on a tour does not exceed the capacity of the vehicle.

Because real-world situations do not necessarily comply with the standard formulation, many extensions and modifications have been examined. One of the standard extensions is the *VRP with Time Windows* in which each customer has an earliest time and a latest time for acceptance of service, defining an interval in which it may be visited. These time windows can be either soft, in which case a penalty can be paid to allow service outside the interval, or hard, in which case

the customer must be served within its time window. See Cordeau et al. (2002) for more information on this extension.

2.1 Dynamic Vehicle Routing

Dynamic VRPs differ from static VRPs in a number of ways. In Psaraftis (1988) the terms are defined as follows: a vehicle routing problem is *static* if the inputs to the problem do not change, neither during execution of the algorithm that solves it, nor during the execution of the solution; a problem is considered *dynamic* when inputs to the problem become known to the decision maker or are updated concurrently with the determination of the solution (i.e., the set of routes).

2.1.1 Dynamic vs. Static Vehicle Routing Problems

Psaraftis mentions 12 issues where the dynamic version may vary from the static version, each of which is discussed to some detail in his paper:

1) The time dimension is essential.
2) The problem may be open-ended.
3) Future information may be imprecise or unknown.
4) Near-term events are more important.
5) Information update mechanisms are essential.
6) Resequencing and reassignment decisions may be warranted.
7) Faster computation times are necessary.
8) Indefinite deferment mechanisms are essential.
9) The objective function may be different.
10) The time constraints may be different.
11) The flexibility to vary fleet size is lower.
12) Queuing considerations may become important.

Psaraftis (1995) elaborates on his definition, supplying some problems that are not dynamic in the sense described above. His examples of static problems include the *Time-Dependent Travelling Salesman Problem*, the *Probabilistic Travelling Salesman Problem* and the *VRP with Stochastic Travel Times*. In the *Time-Dependent Travelling Salesman Problem* the travel times (costs) between customers are 'dynamic' in the sense that they change as a function of time. However, the changes follow a known pattern, and an optimal solution can be found *a priori*; thus one must regard this as a static problem. In the *Probabilistic Travelling Salesman Problem* each customer has a probability p of having a demand, and a route of minimal expected travel time is determined through all customers. Before the route is travelled, the present customers are revealed and the predefined route is followed, except that customers with no demand are skipped. Here, an optimal solution can be determined on the basis of the initial information, i.e., the travel times and probability p, and the final route can be settled by observing actual demand before it is executed.

Psaraftis's final example, the *VRP with Stochastic Travel Times*, belongs to the class of SVRPs discussed in Section 2.2. Even though the travel times and on-site service times are stochastic and

revealed during execution, no decisions are made after the determination of a set of preplanned routes that minimises the given objective (see Kenyon and Morton, 2003).

Psaraftis states as a general rule that the problem is static whenever the output of a certain formulation is a set of preplanned routes that are not reoptimised and are computed from inputs that do not evolve in real-time. A problem is dynamic when the output is not a set of routes, but rather a policy that prescribes how the routes should evolve in time as a function of the inputs.

For a survey on applications of dynamic routing and dispatching, see Gendreau and Potvin (1998). Here the authors divide this broad class of problems by focusing on the dispatching area (local area versus wide area), on the importance of routing, on whether or not there are demands, and on whether it is a many-to-many problem or a one-to-many problem. A many-to-many problem is a problem where each request includes a pick-up location and a delivery location, whereas a one-to-many problem means that each request include a single pick-up (or delivery) location. The authors also note the apparent lack of focus on demand forecasting in contemporary research, which could have a positive impact on the construction of planned routes.

2.1.2 Exploiting information about future events

Whenever a DVRP is at hand, some properties of the future events are usually known. Of utmost importance is the quality of information, that is, whether the information is known deterministically, as a forecast, probabilistically, or unknown altogether. The question addressed in Ichoua, Gendreau and Potvin (2001) is how to exploit knowledge about future demands. The authors discuss two prospects regarding dynamic VRPs: analytical studies and stochastic approaches.

Analytical studies examine the robustness and asymptotic behaviour of different deterministic dispatching strategies (policies) given probability distributions of the random variables. Bertsimas and van Ryzin (1991, 1993) perform such analyses for the *Dynamic Travelling Repairman Problem*, where vehicles are used to service demands independently and uniformly distributed according to a Poisson process. The goal is to find a routing policy that minimises the expected time spent in the system by each customer. Several myopic policies are considered and compared to lower bounds on average system time.

Stochastic approaches, on the other hand, incorporate information about future events in the solution frameworks, in order to produce better solutions than those obtained by myopic, deterministic approaches. One option is to relocate empty vehicles in anticipation of future demands, which is pursued in Larsen (2000) for a *Dynamic Travelling Salesman Problem*. Different strategies are tested for repositioning a vehicle to predefined resting locations (idle points), though the empirical tests did not identify any dominant policy.

Ichoua, Gendreau and Potvin (2001) extend a previous solution approach for a complex *DVRP with Time Windows*. This problem consists of serving customers with soft time windows at minimum cost in the presence of a hard upper bound on the time window of the depot. The cost of a route is dependent on the lateness penalties and on the total distance travelled. The original algorithm, a parallel tabu search heuristic with adaptive memory, is extended by allowing a vehicle to wait at its current position if the probability of new, nearby requests is high. Computational results indicate that the proposed policy is effective.

A methodology that captures the need to take future events into consideration, even when the events are not stochastically modelled or predicted, is presented in Mitrović-Minić, Krishnamurti

and Laporte (2003). The authors suggest using double-horizon based heuristics, where one tries to capture the impact of decisions both on a short-term and a long-term basis. They present computational results for the uncapacitated *Dynamic Pickup and Delivery Problem with Time Windows*.

2.2 Stochastic Vehicle Routing

The SVRP arises when some elements of the problem are stochastic, that is, the value of the element is known only by its probability distribution. This could apply to several of the components of the standard VRP, for instance travel times, demands, or presence of customers. SVRPs are usually formulated as stochastic programming problems, though other possibilities exist, such as Markov decision processes (see, e.g., Dror, Laporte and Trudeau, 1989).

The SVRP comes in many variants, six of which are discussed in Gendreau, Laporte and Séguin (1996a): the *VRP with Stochastic Demands*, the *VRP with Stochastic Customers*, and the *VRP with Stochastic Customers and Demands*, as well as three variants of the *Stochastic Travelling Salesman Problem*, which are of lesser interest here, as they are uncapacitated.

The *VRP with Stochastic Demands* is the most studied SVRP: it is a VRP where customer demands are random variables that become known when the customer is first visited. When stated as a two-stage stochastic program with recourse or a chance constrained program, the *VRP with Stochastic Demands* is a static problem. Different recourse actions can be introduced. Gendreau, Laporte and Séguin (1995) and Laporte, Louveaux and Van hamme (2002) describe exact algorithms for the recourse model where the penalty function is the cost of back and forth trips to the depot due to route failures. A discussion about an alternative characterisation of the *VRP with Stochastic Demands* as a multi-stage problem (see Dror, Laporte and Trudeau, 1989) brings up the issue of reoptimisation, as new information is learned at each stage, i.e., whenever a vehicle arrives at a customer location. The authors also present a Markov decision process formulation that allows choices to be made during the execution of the routes – whether or not to replenish a visited customer – but no computational results are presented in this dynamic setting.

An important case of the SVRP arises when both presence of customers and their demand are stochastic variables, yielding the *VRP with Stochastic Customers and Demands*. It is still a static problem; absent customers are skipped and whenever capacity is exceeded the only possibility is to return to the depot to unload before recommencing service at the last visited customer. As noted by Gendreau, Laporte and Séguin (1995), in this case even the evaluation of the objective function is hard since it calls for the computation of the expected recourse cost. The algorithm described by these authors can only solve relatively small instances to optimality. One conclusion is that stochastic customers are a far more complicating factor than are stochastic demands. Gendreau, Laporte and Séguin (1996b) present a tabu search heuristic for the same problem with promising results.

Because two-stage stochastic programming formulations typically call for the determination of a preplanned set of routes and do not allow for reoptimisation, they are, in the sense discussed in Section 2.1, static problems. However, if a problem is formulated as either a multi-stage stochastic program or a Markov decision process, new information may be revealed at some point, so that reoptimisation becomes an appealing option.

3 The Linjegods case study

The problem discussed in this paper is based on a real-world case observed in one of the leading distribution companies in Norway, Linjegods AS, which has an annual turnover of approximately two billion NOK (as of 2002) and employs about 1150 people. One of the company's depots is located near Trondheim and is used to service both pick-up customers (requesting goods to be fetched at their address and brought to the depot) and delivery customers (needing some goods to be transported from the depot to their location).

For delivery customers the goods arrive at the depot during the night before delivery, and all demand is known when vehicle routes are planned the following morning. Pick-up orders, however, may be called in during the day while other customers are already being serviced, thus making it more difficult to incorporate these in the vehicle routes. Possible recourse actions to enable service of a new customer may be to rearrange customers on existing routes, to reassign customers to other routes or, as is often done in practice, to start an additional vehicle route. Because separate vehicles are used for delivery and pick-up orders, one can view the servicing of the two order types as two distinct problems. Here we will focus on the pick-up part only.

Real-world data for the particular situation depicted above were collected, for other purposes, in a project conducted at SINTEF (Wahl et al., 2002). Customer data were collected for three months, out of which a four-week period contained data of adequate quality. There are on average about 130 pick-up orders each day, distributed geographically in an area of approximately 11 by 13 kilometres. Figure 1 illustrates the different locations of pick-up orders during four weeks of operation. The coordinates of the depot are roughly (500,1500) as shown by the larger square.

Although many of the customer locations are frequently visited – some addresses had more than 200 registered requests over the four-week interval of data examined – many required only a few visits; about half of all registered addresses had only requested one pick-up.

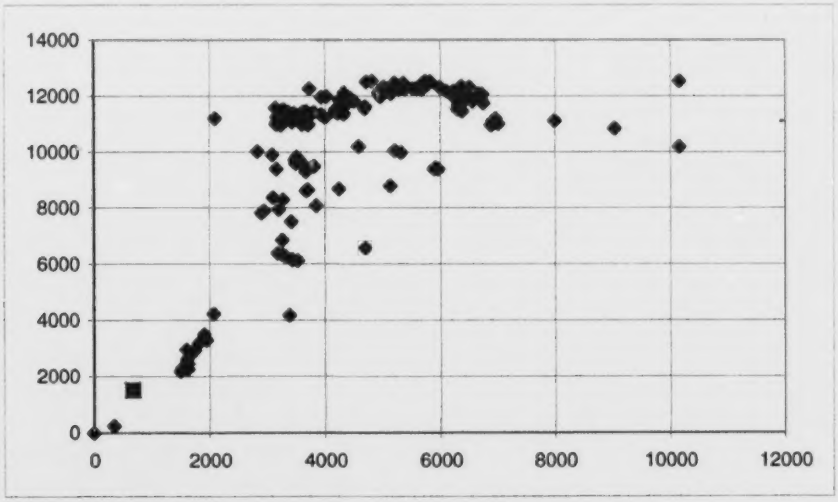


Figure 1: Locations of pick-up customers in the Linjegods case study during four weeks of operation

Unfortunately, the available data do not include information on call-in time for the customers. The profiles of customer calls are available, however, from other booking offices of the same company. Assuming these are representative of the Trondheim depot, we can use the distribution of Figure 2. Around 50% of the customers have registered their orders before 07:00, and are thus known when initially planning the vehicle routes for a particular day. The remaining orders are reported to the booking office between 08:00 and 14:00. Other attributes of the orders include the weight and volume of the goods, the time interval for which service is acceptable, and the duration of the service at the location of the customer.

Distributions are calculated for the different attributes, based on the collected data, so that random problem instances may be generated. The distributions are generated as follows. Rather than using distances based on a real road network, we use Euclidian distances. The geographical area is divided into $n \times n$ sectors, and customer locations are assumed to be uniformly distributed within each sector. Historical customer locations (Figure 1) are then used to find the probability that a customer shows up in a given sector. The Poisson distribution is used to represent the number of customers appearing at each time interval, as is typically done in the literature (see, for instance, Larsen, 2000). One disadvantage of the call-in profile being separated from the customer location data is that any dependence between the two is lost. That is, if customers in one geographical area are more likely to call in their requests in one particular time interval, this information, which could affect the solution strategy, will disappear. Hence, in this work, the same geographical distribution is used regardless of the time of day at which the order becomes known. See Ichoua, Gendreau and Potvin (2001) for a contrasting example.

Another issue with respect to the time dimension is the distribution of time windows. From the raw data it is clear that the depot is open from 07:00 to 18:00, and although information about time windows is missing for most of the pick-up customers, the only specified time windows for customers are also from 07:00 to 18:00. Therefore we assume that the time windows are in general very wide, and correspondingly we use the call-in time and 18:00 as start and end of the customer time windows.

Regarding capacity constraints, only the weight is considered here. Rather than fitting the observed data to a well-known statistical distribution we assign each registered weight in the historical database an equal probability of reappearance as the demand of a new customer.

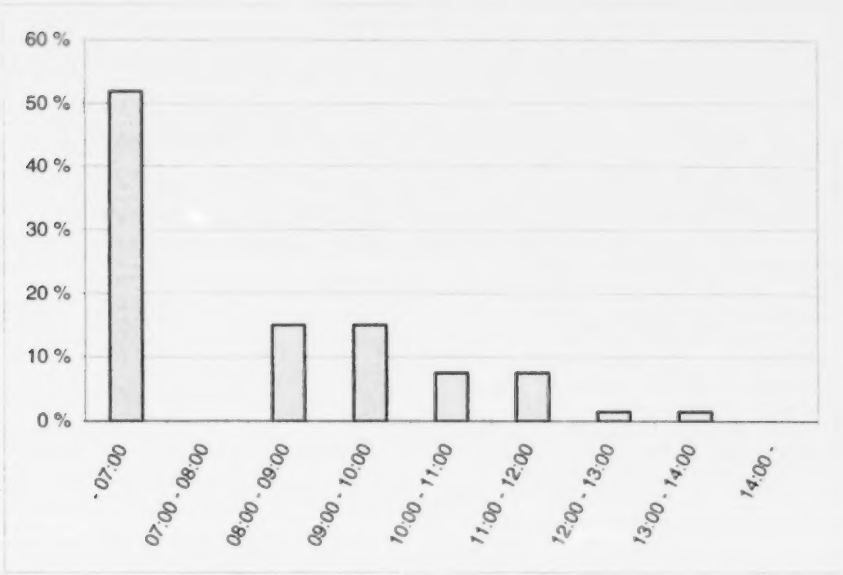


Figure 2: Distribution of call-in time of orders in the Linjegods case study

4 Mathematical problem description

The problem described in the previous section combines aspects of the DVRP and of the SVRP, and does not seem to fit within any existing formulation found in the literature. Because customers can call in orders during the entire working day, the problem must be handled as a dynamic problem. The presence of historical data, however, suggests that a stochastic solution strategy is appropriate, i.e., information about distributions of unknown variables could be advantageously used. At least, Figure 1 indicates that the geographical spread of customers should be considered when planning initial routes. Also, the fact that the frequency of customer appearances is time-dependent (see Figure 2), implies that an optimal routing policy may change during the day.

The location and time window of a customer become known at the time when service is accepted. Each order has a weight, and the vehicle that services the customer must have sufficient capacity to accommodate the pick-up load. For deterministic customers we assume that the time window and the weight of the order are known, whereas the number of stochastic customers, their call-in time, as well as the location, time window, and demand of each individual stochastic customer, are only known probabilistically. The location, time window and demand will, however, be revealed at call-in time. To handle the appearing stochastic customers some recourse actions must be defined. We allow that customers already in the planned vehicle routes may be rearranged within their vehicle route or reassigned to a different vehicle route, and that new vehicle routes may be started. However, preemption of customers is not accepted: if a vehicle has already left its previous customer, then the next customer on its route must be visited immediately, without detours.

4.1 Deterministic formulation of the VRP with time windows

In order to better appreciate the mixed integer program formulation of the stochastic problem, it is useful to first provide the formulation of the deterministic problem, based on earlier classical formulations (see, e.g., Cordeau et al., 2002). The deterministic capacitated VRP with time windows and a homogeneous fleet of vehicles can thus be formulated using the following parameters and variables.

Let the set of customers be denoted by $C = \{1,2,...,n\}$ and let 0 be the depot, so that the set of all locations is $N = C \cup \{0\}$. Travel times c_{ij} are given between every pair of locations $i, j \in N$, and include service time. Every customer i has a demand d_i and a time window $[a_i, b_i]$ for service. The time window for the depot is given by $[a_0, b_0]$ and, like the time windows for the customers, is considered hard, i.e., service must start after a_i and before b_i. (Since service time is included in c_{ij}, it must also be subtracted from the start and end of the corresponding time window.) A set of vehicles, $V = \{1,2,...,k\}$, each of capacity q, start at the depot at time a_0 and must be back at time b_0, after servicing the customers.

The decision variables of the model are x_{ijk}, where $x_{ijk} = 1$ if vehicle k travels directly from i to j, and $x_{ijk} = 0$ otherwise. In addition, s_{ik} denotes the time at which vehicle k has completed service at customer i. Note that the variables s_{ik} make designated subtour elimination constraints redundant. The formulation is then:

$$\min\ z(x,s) = \sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk} + K\sum_{k\in V}(1 - x_{00k}) \qquad (1)$$

s.t.

$$\sum_{k\in V}\sum_{i\in N} x_{ijk} = 1 \qquad \forall j \in C \qquad (2)$$

$$\sum_{j\in N} x_{0jk} = 1 \qquad \forall k \in V \qquad (3)$$

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0 \qquad \forall h \in C, \forall k \in V \qquad (4)$$

$$\sum_{i\in N} x_{i0k} = 1 \qquad \forall k \in V \qquad (5)$$

$$\sum_{i\in C} d_i \sum_{j\in N} x_{ijk} \le q \qquad \forall k \in V \qquad (6)$$

$$s_{ik} + c_{ij} - M_{ij}(1 - x_{ijk}) \le s_{jk} \qquad \forall j \in C, \forall i \in N, \forall k \in V \qquad (7)$$

$$a_i \le s_{ik} \le b_i \qquad \forall i \in N, \forall k \in V \qquad (8)$$

$$s_{ik} + c_{i0} - M_{i0}(1 - x_{i0k}) \le b_0 \qquad \forall i \in C, \forall k \in V \qquad (9)$$

$$s_{0k} = a_0 \qquad \forall k \in V \qquad (10)$$

$$x_{iik} = 0 \qquad \forall i \in C, \forall k \in V \qquad (11)$$

$$x_{ijk} \in \{0,1\} \qquad \forall i, j \in N, \forall k \in V. \qquad (12)$$

Here, the objective (1) is to minimize the number of vehicles used plus the total travel time. The constant K represents the cost of using an extra vehicle, and is sufficiently large to ensure that the number of vehicles is always more important than the total distance travelled. Constraints (3), (4) and (5) state that each vehicle should leave the depot once, leave each customer at which it arrives, and return to the depot once. Inequalities (6) guarantee that vehicle capacity is not violated. Constraints (7) ensure that vehicle k spends the correct amount of time driving between i and j before servicing customer j, and inequalities (8) define the time window of each customer. Finally constraints (9) force every vehicle to arrive at the depot before the end of the depot's time window. The constant M_{ij} is defined to be sufficiently large, e.g., $M_{ij} = b_i + c_{ij} - a_j$.

The model above is deterministic, that is, it contains no random variables. Since our problem contains stochastic elements (i.e., variables whose probability distributions are known) and a dynamic element (i.e., information becomes known throughout the day, and replanning is warranted), the model must be extended.

4.2 A two-stage stochastic model

In order to capture the stochastic elements of the problem, we extend the deterministic model to a two-stage stochastic model. Here, we assume that there exists a single known moment $t \in [a_0, b_0]$, at which all previously unknown information is revealed. That is, all stochastic customers, their location, demand and time windows become known at time t. At this moment recourse actions may be considered, so that the new customers may be included in the vehicle routes.

Extending the deterministic model of Section 4.1, we introduce new variables to handle recourse actions. As before, $C = \{1,2,...,n\}$ is the set of initially known customers, and 0 is the depot. Let $C^t = \{n+1, n+2, ..., n+n^t\}$ be the set of customers revealed at time t. Let d_i and $[a_i, b_i]$ be the customer demand and time window, $i \in C^t$. Let the set of all locations be $N' = C' \cup \{0\}$, where $C' = C \cup C^t$. The travel time between i and j is c_{ij}, $i, j \in N'$. Let $\xi = (C^t, D^t, A^t, B^t, c^t)$ be a particular realization of the random variable vector $\tilde{\xi} = (\tilde{C}^t, \tilde{D}^t, \tilde{A}^t, \tilde{B}^t, \tilde{c}^t)$. Then the two-stage stochastic programming problem can be formulated as

$$\min \; E_{\tilde{\xi}}\left[z(x,s) + Q(x,s,\tilde{\xi})\right] \qquad (13)$$

$$Q(x,s,\tilde{\xi}) = \sum_{i \in N'} \sum_{j \in N'} \sum_{k \in V} c_{ij} \left(x_{ijk}^{t+} - x_{ijk}^{t-}\right) + K \sum_{k \in V} \left(x_{00k}^{t-} - x_{00k}^{t+}\right) \qquad (14)$$

s.t. (2)-(12) and

$$\sum_{k\in V}\sum_{i\in N'}\left(x_{ijk}+x^{t+}_{ijk}-x^{t-}_{ijk}\right)=1 \qquad \forall j\in C' \qquad (15)$$

$$\sum_{j\in N'}\left(x_{0jk}+x^{t+}_{0jk}-x^{t-}_{0jk}\right)=1 \qquad \forall k\in V \qquad (16)$$

$$\sum_{i\in N'}\left(x_{ihk}+x^{t+}_{ihk}-x^{t-}_{ihk}\right)-\sum_{j\in N'}\left(x_{hjk}+x^{t+}_{hjk}-x^{t-}_{hjk}\right)=0 \qquad \forall h\in C',\forall k\in V \qquad (17)$$

$$\sum_{i\in N'}\left(x_{i0k}+x^{t+}_{i0k}-x^{t-}_{i0k}\right)=1 \qquad \forall k\in V \qquad (18)$$

$$\sum_{i\in C} d_i \sum_{j\in N}(x_{ijk}+x^{t+}_{ijk}-x^{t-}_{ijk})\le q \qquad \forall k\in V \qquad (19)$$

$$x^{t-}_{ijk}\le x_{ijk} \qquad \forall i,j\in N,\forall k\in V \qquad (20)$$

$$x^{t-}_{ijk}=0 \qquad \forall i,j\in C',\forall k\in V \qquad (21)$$

$$tx^{t-}_{ijk}\le s_{ik} \qquad \forall i,j\in N,\forall k\in V \qquad (22)$$

$$s^t_{ik}+c_{ij}-M_{ij}(1-x_{ijk}-x^{t+}_{ijk}+x^{t-}_{ijk})\le s^t_{jk} \qquad \forall j\in C',\forall i\in N',\forall k\in V \qquad (23)$$

$$a_i\le s^t_{ik}\le b_i \qquad \forall i\in N',\forall k\in V \qquad (24)$$

$$s^t_{ik}+c_{i0}-M_{i0}(1-x_{i0k}-x^{t+}_{i0k}+x^{t-}_{i0k})\le b_0 \qquad \forall i\in C',\forall k\in V \qquad (25)$$

$$s^t_{0k}=a_0+(t-a_0)x_{00k} \qquad \forall k\in V \qquad (26)$$

$$x^{t+}_{iik}=0 \qquad \forall i\in C',\forall k\in V \qquad (27)$$

$$x^{t+}_{ijk}\in\{0,1\} \qquad \forall i,j\in N',\forall k\in V \qquad (28)$$

$$x^{t-}_{ijk}\in\{0,1\} \qquad \forall i,j\in N',\forall k\in V. \qquad (29)$$

Here, x^{t+}_{ijk}, x^{t-}_{ijk} and s^t_{ik} depend on the random vector $\tilde{\xi}$, and are thus random variables themselves. The expected value with respect to the distribution of $\tilde{\xi}$ is denoted by $E_{\tilde{\xi}}$. The random variable $Q(x,s,\tilde{\xi})$ is the recourse cost, and sums up the cost of starting new vehicles and the change in travel time. The variable $x^{t+}_{ijk}=1$ if vehicle k travels directly from i to j in second-stage recourse but not in first-stage solution, and $x^{t-}_{ijk}=1$ if vehicle k travelled directly from i to j in first-stage solution but not after recourse. The variable s^t_{ik} represents the time at which vehicle k has completed service at customer i according to the second-stage recourse actions.

Constraints (15)-(18) are similar to (2)-(5), ensuring that all customers are served, that any vehicle must leave the depot once, leave all customers that are visited, and arrive at the depot once. Constraints (19) ensure that vehicle capacities are not violated in second-stage solution.

Constraints (20)-(22) state that a trip between customer i and j in the first-stage solution cannot be skipped in second-stage recourse if service at departure-location is completed before time t. A vehicle k that does not leave the depot in the first-stage solution ($x_{00k} = 1$) cannot leave the depot before t in a second-stage decision, as stated by (26). Constraints (23)-(25) are used to track the time at which service is completed in the final, second-stage solution.

A challenge, from a modelling perspective, with the given two-stage formulation is that the number of new customer requests arriving up to point t, i.e., $|C^t|$, is also a random variable. To remedy this, one can define an upper limit L on the number of new requests, so that the probability $\Pr(0 \le |C^t| \le L)$ of having between 0 and L new requests is 1. In the formulation, let $|C^t| = L$, and assume that the random variables corresponding to customers that do not appear have a negligible impact. That is, if $i \in C^t$ represents a customer that does not appear, then $c_{ij} = 0$, $d_i = 0$, and $[a_i, b_i] = [a_0, b_0]$. The constraints and objective function are then well defined *a priori*, and the probability distributions of the random variables may be suitably modified to accommodate the changes.

Constraints (15)-(26) in the above stochastic formulation limit the available recourse and force the values on some of the variables in the second-stage solution depending on the first-stage solution. Under these constraints, one is allowed to completely modify the routes after new information has been revealed at time t, except that past events cannot be undone (thus the solution policy is *implementable*, see Rockafellar and Wets, 1991). That is, customers not visited before time t at which the next stage begins may be rearranged or even reassigned to a different vehicle.

Whenever, in a two-stage formulation, the feasibility of the first-stage solution implies that the second-stage problem is feasible for any realisation of the stochastic variables, the problem is said to have *relatively complete recourse*. This attribute is unfortunately not present here, as there may appear customers in the second stage that are, as a consequence of first stage decisions, impossible to serve within their time windows. Since the constraints state that every customer must be visited exactly once, either the time related constraints (22)-(26) or the service related constraints (15)-(19) may be violated.

However, the property of relatively complete recourse may be enacted, if one is allowed to skip service of customers at the expense of some high penalty. Let A be the penalty per unserviced customer, and let a_j, $j \in C'$, be new decision variables such that $a_j = 0$ if customer j is serviced, and $a_j = 1$ if the customer is not serviced. In the two-stage formulation, the new recourse cost component would be:

$$Q(x, a, s, \tilde{\xi}) = \sum_{i \in N^t} \sum_{j \in N^t} \sum_{k \in V} c_{ij} \left(x_{ijk}^{t+} - x_{ijk}^{t-} \right) + K \sum_{k \in V} (x_{00k}^{t-} - x_{00k}^{t+}) + A \sum_{j \in C^t} a_j . \qquad (14b)$$

The constraint that ensures that all customers are visited is changed as follows:

$$\sum_{k \in V} \sum_{i \in N^+} \left(x_{ijk} + x_{ijk}^{t+} - x_{ijk}^{t-} \right) + a_j = 1 \qquad \forall j \in C^+ \qquad (15b)$$

$$a_j \in \{0,1\} \qquad \forall j \in C^+. \qquad (30)$$

The value of A would typically be set very high so that one would never voluntarily choose not to service a customer. This approach could be extended to a multi-stage formulation, allowing customers to be unserviced in the final execution and making relatively complete recourse valid.

The recourse actions implied by the above stochastic model are rather liberal compared with those normally described in the literature. In order to efficiently compute the objective function, including the expected recourse cost, the recourse actions should not be very intricate. Even when the recourse is kept simple, such as inserting return trips to the depot in case of capacity violations and otherwise following the planned routes, computing the expected recourse cost for a particular solution may be exceedingly difficult (see Gendreau, Laporte and Séguin, 1995, 1996b). Ahmed and Shapiro (2002) discuss the difficulties related with recourse cost evaluation for general stochastic programs with integer recourse. Obtaining the exact value of the expected recourse costs for a two-stage stochastic program requires multi-dimensional integration of integer programming value functions if continuous distributions are used, and for uncertain parameters with discrete distributions, recourse evaluations would require solving the integer recourse problem for all possible realizations of the variables.

4.3 A multi-stage stochastic model

As stated, the call-in time for each stochastic customer is unknown. Thus, the two-stage model just described does not fully capture the dynamic element of the real-world case upon which it is based. A possible solution would be to treat the number of stages in the problem as a stochastic variable, with one stage per call-in. In this work, however, we opt for a simpler approach in which we divide the time-horizon into a prespecified number v of intervals. The objective is then to find a plan for the remainder of the day, using the currently known customers. This is repeated dynamically for every time-interval, each time modifying the problem with information received during the previous interval. By including additional variables and constraints for each stage, it is straightforward to expand the given two-stage model into an m-stage model for any given m. Since the procedure for extending the model to m stages is akin to extending the deterministic model to a two-stage model, we omit the details of this multi-stage formulation, and direct the reader to Kall and Wallace (1994) for more on stochastic formulations.

The multi-stage stochastic model now captures both the stochastic and dynamic elements of the real-world case from Section 3. For each stage a solution must be produced, taking into account both the information revealed up to the corresponding moment in time as well as the stochastic information about future events.

5 The Dynamic Stochastic Hedging Heuristic

Exact solution methods for two-stage stochastic VRPs currently fail to consistently solve problems with more than just a few customers (see Gendreau, Laporte and Séguin, 1995 and 1996b) unless additional assumptions on the shape of the solutions are made (Laporte, Louveaux and Van hamme, 2002). In our case, even the evaluation of the recourse cost function (14) can be

extremely difficult, depending on the distribution of the random variables. Hence, the need for practical heuristic solution methods is evident.

Rather than letting the solution method depend on assumptions about the distributions of the stochastic variables, we have developed a framework based on sample information, an idea that can be traced back to Jagannathan (1985). Sample information can often be used in solution strategies that produce optimal solutions with probability approaching one as sample size is increased, or in bounding techniques to validate the quality of a candidate solution (see Ahmed and Shapiro, 2002). For two-stage stochastic routing problems Verweij et al. (2003) have produced near-optimal solutions by solving the sample average approximation problem, which is minimisation over the first stage cost plus the corresponding average recourse cost of a set of sample scenarios.

In the approach presented here, sample scenarios are used to guide a heuristic method that builds a plan for each time interval in turn. The aim is to solve each of the sample scenarios as static VRPs, and then use common features from the solutions to build a hopefully good plan. The idea of building the solution based on shared features in sample solutions is partly rooted in progressive hedging as found in Løkketangen and Woodruff (1996), being first introduced by Rockafellar and Wets (1991). Hence, the new heuristic approach will be called the Dynamic Stochastic Hedging Heuristic (DSHH).

The algorithm assumes that random scenarios can be generated, all having the same deterministic customers (those known at the current time), and having stochastic customers whose attributes are drawn from the correct distributions. Thus, either the real distributions or some approximated distributions must be available.

We now state the DSHH. Assume the time horizon is divided into v intervals, $I_1,...,I_v$, which corresponds to stages in the multi-stage model. That is, at the beginning of each interval the goal is to determine a plan that minimizes the expected cost of servicing all customers, both future and currently known. In a sense, the heuristic has a dynamic part (the replanning occurring at the start of each interval) and a stochastic part (the use of stochastic information during planning). The outer loop of the heuristic, which concerns the dynamic aspect, is described in Box 1.

A. At the start of each interval $I_1,...,I_v$:
   a. Take the solution from the previous interval and lock the plan up to the point in time where the current time interval begins. If a vehicle has left its last customer, a lock is set to indicate that it must continue to the customer that followed in the solution from the previous interval.
   b. Add to the problem all customer requests that became known during the last interval.
   c. Find plans (for the current interval, I_u) by using the DSHH subprocedure.

B. Evaluate the final solution.

Box 1: Outer loop of DSHH

Observe that we exploit the fact that the plan for the intervals $I_{u+1},...,I_v$ may change at the start of interval I_{u+1} (as a result of recourse actions), by allowing the algorithm not to explicitly create a plan for more than the very next interval at each repetition. Although the algorithm could be extended to produce a plan for the entire set of subsequent time intervals, this is deemed redundant in view of the anticipated recourse actions. The four phases of the DSHH subprocedure, which enclose the stochastic part of the heuristic, are defined in Box 2.

Let C be the set of known customers at the start of I_u, and let S be a subset of C, initially empty.

1. Create r sample scenarios with the same known customers as in C, including stochastic customers drawn from the given distributions.
2. Repeat the following:
   2.1. Solve the sample scenarios, while forcing all customers in S to be served during I_u.
   2.2. Find the customer $c \in C - S$ that is most frequently visited in the time interval I_u in the r sample scenarios.
   2.3. If a stopping criterion is met then go to 3, otherwise let $S := S \cup \{c\}$.
3. Repeat until S is empty:
   3.1. Solve the sample scenarios.
   3.2. Count the number of times each customer $c \in S$ is served first by each vehicle in each scenario solution, disregarding customers that have already been placed.
   3.3. Choose the customer-vehicle pair with the highest count. Lock the customer to be served before all remaining customers by the given vehicle, and remove it from S.
4. Terminate with a plan for the given time interval I_u.

Box 2: The DSHH subprocedure

After each repetition of the DSHH subprocedure customers served during the previous time intervals can be ignored in the solution process, as long as the position and spare capacity of each vehicle is remembered. We now detail the first three phases of the DSHH subprocedure.

5.1 Phase 1 of the DSHH subprocedure

In phase 1 of the DSHH subprocedure, knowledge about the probability distributions of the stochastic variables of the model, i.e., distributions of call-in times, demands, geographical locations, etc., is used to create possible future events. Our sample scenarios are obtained by combining randomly selected future events with the known customers and their attributes. The sample scenarios have the property that likely events are present with a high probability.

5.2 Phase 2 of the DSHH subprocedure

The second phase identifies which customers should be included in the routes for the current time interval. First, solutions of the sample scenarios are found using a simple insertion heuristic, as if they were deterministic VRPs: start with the routes corresponding to the plan already executed. The parts of the routes already executed are handled implicitly, i.e., the sample scenarios are

modelled as a *VRP with Time Windows*, with heterogeneous vehicles and different start locations for each vehicle. Each start-depot matches the current position of a vehicle, and the capacity of each vehicle has been reduced to account for the goods collected up to the present time. The end-depot is the same for all vehicles, and corresponds to the real depot. Time windows are also appropriately modified. Customers included in the sample scenario are those currently known but unserviced, as well as the randomly sampled future requests. A sample scenario solution is completed by sequentially inserting customers into the routes: for each customer, try in turn all possible legal insertion points in the routes and select the point yielding the least possible cost. The order in which customers are handled has influence on the end result, but is not actively controlled by the insertion heuristic, and might therefore in some sense be considered random.

As the second phase continues, customers that are frequently served in the time interval under consideration are identified, and a decision is made to visit these customers during I_u in the final plan. Sample scenarios where these customers were not already served in I_u are then solved again, enforcing the decision in all sample scenario solutions, before the selection procedure continues. The selection is considered complete when some stopping criterion is reached: 1) There are no unselected customers that are frequently handled during I_u in solutions to the sample scenarios. A customer is considered infrequent if it is served during I_u in less than p_1% of the r solutions. 2) A sufficient number of customers have been selected to occupy a given fraction of the vehicles for the duration of I_u. This fraction is measured relative to the minimum number of vehicles used in the sample scenario solutions.

To speed up the second phase, more than one customer are allowed to be included in the set S at each repetition. This is achieved by including additional customers if they are served during I_u in more than p_2% of the r solutions, where $p_2 \geq p_1$. If none of the stopping criteria is triggered, phase 2 is then repeated. Note that decisions made regarding the customers in S will also be enforced in the sample scenario solutions, so that decisions made in each repetition will depend on previous decisions.

After phase 2 is completed and the customers that should be served in the current time interval are determined one still has to decide in which sequence they should be visited. The information contained in the sample scenario solutions is again exploited in this process.

5.3 Phase 3 of the DSHH subprocedure

The third phase corresponds to finding the actual routes with the customers selected in phase 2. After solving the sample scenarios with the customers of S locked in I_u, count how many times each customer is served first on each vehicle while disregarding all customers not in S. The customer-vehicle combination with the highest count is chosen and locked in the corresponding position, as well as being removed from S. This selection is repeated until S is empty, at which point all vehicle routes are completed. In this fashion, routes for each of the time intervals may be constructed sequentially based on known information and distributions of the random variables.

5.4 Description of alternative solution methods

Before results and analyses are presented in Section 6, we describe two alternative solution methods, used for comparison purposes. One is based on the solution of the corresponding

deterministic model in Section 4.1, while the other is based on the solution of a pure dynamic problem in which stochastic information is ignored. The former method produces solutions as if all information were known from the beginning of the day by using a local search heuristic. This can be used to compute the value of perfect information as in Mitrović-Minić, Krishnamurti and Laporte (2003). The latter is called the Myopic Dynamic Heuristic (MDH), which at each stage solves the problem using known information only by using the local search heuristic used to solve the deterministic model. Still assuming that the time horizon is divided in v intervals, $I_1,...,I_v$, the MDH is described in Box 3.

1. Heuristically solve a VRP with the initially known customers.
2. At the start of each interval $I_2,...,I_v$:
    2.1. Take the solution from the previous interval and lock the plan up to the moment at which the current time interval begins. If a vehicle has left its last customer, a lock is set to indicate that it must continue to the customer that followed in the solution from the previous interval.
    2.2. Add to the problem the customer requests that have become known during the last interval.
    2.3. Solve the new problem, taking into consideration the locks that have been added.
3. Evaluate the final solution.

Box 3: The MDH

Note that among the three methods mentioned only the MDH and the DSHH are able to produce solutions of the multi-stage formulation. Figure 3 illustrates how real-world events, the planning, and the execution of the plans may evolve during the day with each stage corresponding to one hour.

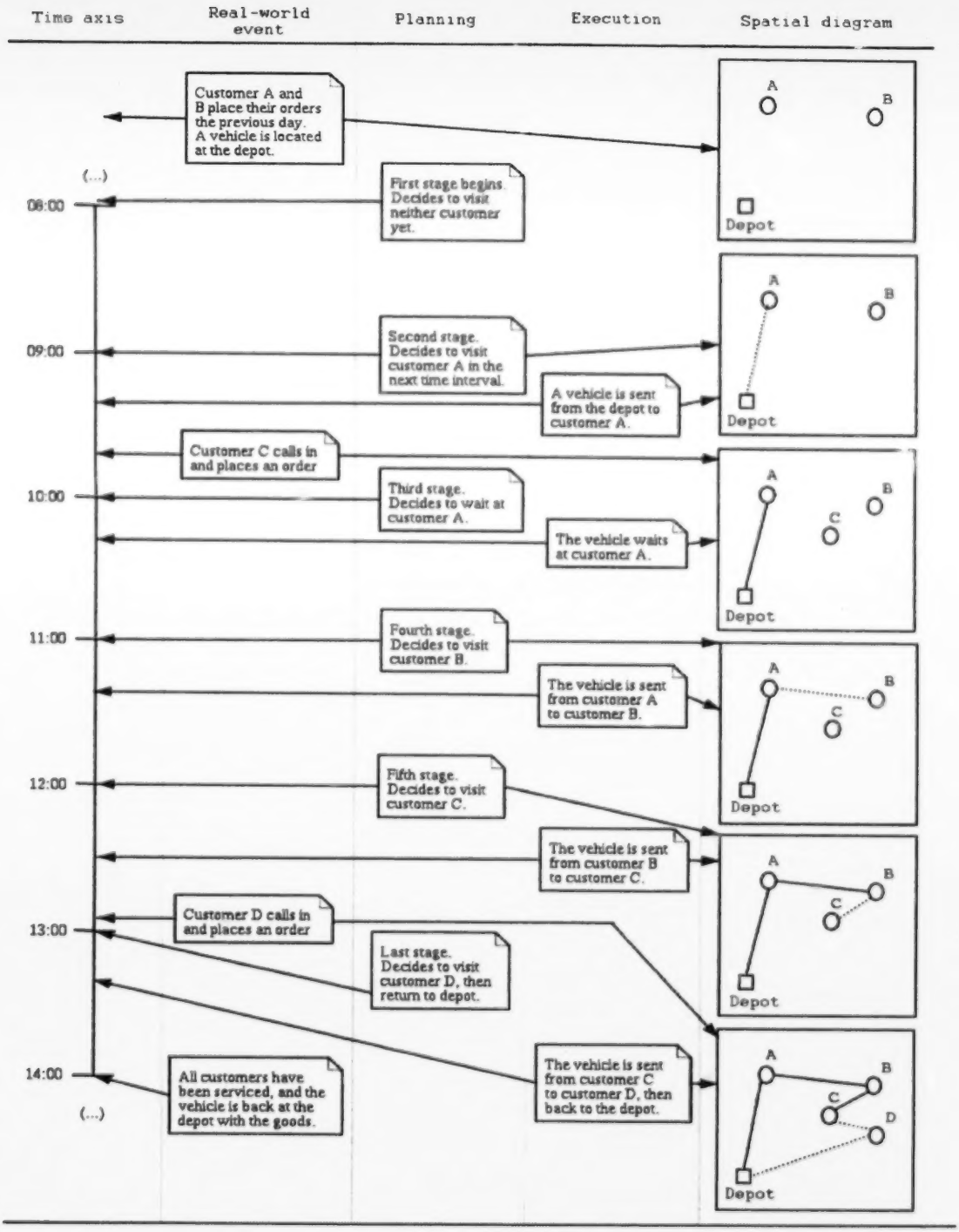

Time axis
Real-world event
Planning
Execution
Spatial diagram
Customer A and B place their orders the previous day. A vehicle is located at the depot.
(...)
08:00
First stage begins. Decides to visit neither customer yet.
09:00
Second stage. Decides to visit customer A in the next time interval.
A vehicle is sent from the depot to customer A.
Customer C calls in and places an order
10:00
Third stage. Decides to wait at customer A.
The vehicle waits at customer A.
11:00
Fourth stage. Decides to visit customer B.
The vehicle is sent from customer A to customer B.
12:00
Fifth stage. Decides to visit customer C.
The vehicle is sent from customer B to customer C.
Customer D calls in and places an order
13:00
Last stage. Decides to visit customer D, then return to depot.
The vehicle is sent from customer C to customer D, then back to the depot.
14:00
All customers have been serviced, and the vehicle is back at the depot with the goods.
(...)
A
B
C
D
Depot

Figure 3: Interaction between real-world events and planning. Dotted lines correspond to planning decisions; solid lines correspond to completed sections of vehicle routes.

6 Results and analyses

The heuristic just described was coded in C++ and run on a Pentium-III 1GHz computer. In order to test the DSHH, 20 test instances were generated based on the distributions outlined in Section 3. The following settings were used during test runs: ten time intervals (i.e., ten stages), each of 3600 seconds, $r = 30$ new sample scenarios were generated and solved for each time interval, and customer selection (phase 2 of the algorithm) also stops if the selected customers fills a number of vehicles equal to the least number of vehicles used in any of the sample scenario solutions. Based on preliminary testing, values for p_1 and p_2 were set to 50% and 70%, respectively.

Results are presented in Table 1 for the 20 test instances, in the column 'DSHH'. Note that in these tests, the heuristic is executed without considering the effect of the elapsed computational time on the actual implementation of the planned routes. That is, we assume that all the information arriving up to the start of a time interval (stage) can be used and that the plan resulting from running the heuristic can be implemented instantly, ignoring the computation time.

For comparison, the results for the deterministic problem (in the column 'Deterministic') and the MDH are also provided in Table 1. The MDH is set to expend approximately the same computational effort as the DSHH, i.e., a total of about six minutes per test instance.

Table 1: Comparative computational results for three algorithmic strategies

		Deterministic		MDH		DSHH		
Instance	No. of orders	Total distance	No. of vehicles	Total distance	No. of vehicles	Total distance	No. of vehicles	Reduction of distance in % with respect to MDH
1	142	147238	6	225570	6	168493	6	25.3%
2	126	111240	4	160817	4	136445	4	15.2%
3	147	144471	5	206618	5	163966	5	20.6%
4	118	122244	4	164989	4	163170	5	1.1%
5	126	134448	5	185090	5	158696	5	14.3%
6	118	113658	4	178188	4	126267	5	29.1%
7	128	124070	5	160181	5	154228	5	3.7%
8	129	149118	5	213679	5	190416	6	10.9%
9	117	97383	3	134465	3	120419	3	10.4%
10	127	138753	5	190106	5	160358	5	15.6%
11	146	125659	4	170566	4	145206	4	14.9%
12	151	121650	4	195716	4	134665	4	31.2%
13	138	104251	3	165455	4	123126	3	25.6%
14	126	122230	4	172262	4	145624	4	15.5%
15	139	128013	5	195932	5	155918	5	20.4%
16	122	117193	4	193146	4	145148	4	24.9%
17	125	143836	5	207379	5	166411	5	19.8%
18	143	155267	6	209961	6	211594	6	-0.8%
19	144	138963	5	187940	5	179210	5	4.6%
20	139	143309	5	181962	5	171903	5	5.5%
Avg.	**132.55**	**129150**	**4.55**	**185001**	**4.60**	**156063**	**4.70**	**15.6%**

A first glance at the results indicates that, compared with the myopic dynamic strategy, the sample-based approach does indeed succeed at reducing the total travel cost. In only one out of 20 instances does the MDH produce shorter routes. However, the DSHH uses on average 4.7 vehicles, as opposed to 4.6 for the dynamic approach. The reduction in distance travelled can be close to 30%, and on average the routes produced by the DSHH is about 15% shorter than the routes produced by the MDH.

In three out of the 20 test runs the DSHH uses one vehicle more than the MDH, and in one run it uses one vehicle less. However, using additional vehicles is not necessarily a bad choice if the probability of needing the extra vehicle becomes high at some point in time. The DSHH only adds an extra vehicle if the majority of the sample scenario solutions need an extra vehicle. In fact, in the three test runs where DSHH uses one more vehicle, the surplus capacity of the vehicles would have been very low if the extra vehicle had not been used. Since the MDH never takes future events into consideration, it will not reserve space on vehicles to accommodate new customers, which in these cases turn out to be the better choice with respect to vehicle usage.

The DSHH results presented in Table 1 come from one particular choice of parameters. By varying the value of some of these parameters one can gain insight into how the method works. In Figure 4 the number r of sample scenarios used in each execution of the DSHH subprocedure (see Box 2) varies between 5 and 150. The travel distance clearly improves as more scenarios are used, and vehicle usage also seems to stabilize, although it seldom becomes smaller than 4.7 on average. Table 2 shows the effect of increasing the sample size for one particular test instance, running the DSHH with ten different random seeds (thus resulting in different sample scenarios being used for each of ten runs on the same test instance). Note from Table 2 that the increase in computational time seems to be linear with respect to the number of sample scenarios. Increasing the number of sample scenarios may thus be an effective way of improving solution quality.

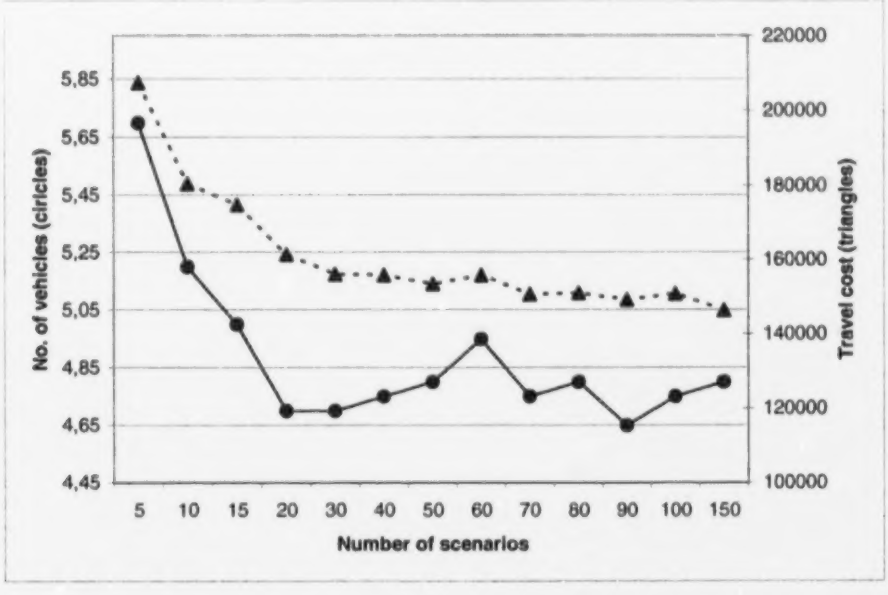


Figure 4: Number of scenarios vs. quality of solution (DSHH)

Table 2: Varying number of scenarios for test instance number 10

No. of samples	No. of vehicles		Total distance		CPU seconds	
	Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
30	5.5	0.53	170072	10412	326	10
100	5.2	0.42	159625	9675	1125	55
300	5.1	0.32	155782	6325	3545	97
600	5.1	0.32	151410	6097	7083	82

Another way of increasing computational effort is to change the number of stages in the model. Rather than updating the plan every hour, one can vary the number of stages (or the number of seconds per interval). Results are depicted in Figure 5. Intuitively, reducing the time length between stages allows the heuristic to react quicker to new information, which in turn should lead to improved solutions. It is also evident that excessively reducing the size of each time interval can cause the heuristic to fail; the hollow circles and triangles in Figure 5 indicate that some customers where left unserviced in some of the instances. The reason why some customers are never included in the plan is linked to the fact that the heuristic only constructs routes for the next time interval. There may be customers that cannot be serviced within 600 seconds in the sample scenario solutions because they are located far from the current position of all the vehicles, and these will never be selected in phase 2.2 of the DSHH.

Thus, when interval sizes are small the heuristic must be modified to consider information from the sample scenario solutions outside of the next interval. Table 3 shows results obtained when allowing the DSHH to look at intervals $I_u,...,I_{u+p}$ rather than just I_u when creating the routes. When each stage in the model corresponds to a 600 second interval, $p = 2$ gives the best results. Higher values of p still allow all customers to be serviced, but increasing the number of customers selected in phase 2 of the DSHH makes the decisions more difficult in phase 3, thus deteriorating the results. Possible remedies could be to spend more effort improving the solutions of the sample scenarios during phase 3 or to somehow be able to determine the correct value of p adaptively, based on the number of customers selected during phase 2.

Table 3: Planning for more than the next interval when using many stages

No. of vehicles	Total distance	CPU seconds	Unserviced	p	Size of $I_u \cup ... \cup I_{u+p}$
5.75	147380	632	20	0	600
5.15	145597	711	5	1	1200
4.75	153656	791	0	2	1800
4.90	162010	865	0	3	2400
4.80	166824	980	0	4	3000
4.90	170202	1121	0	5	3600

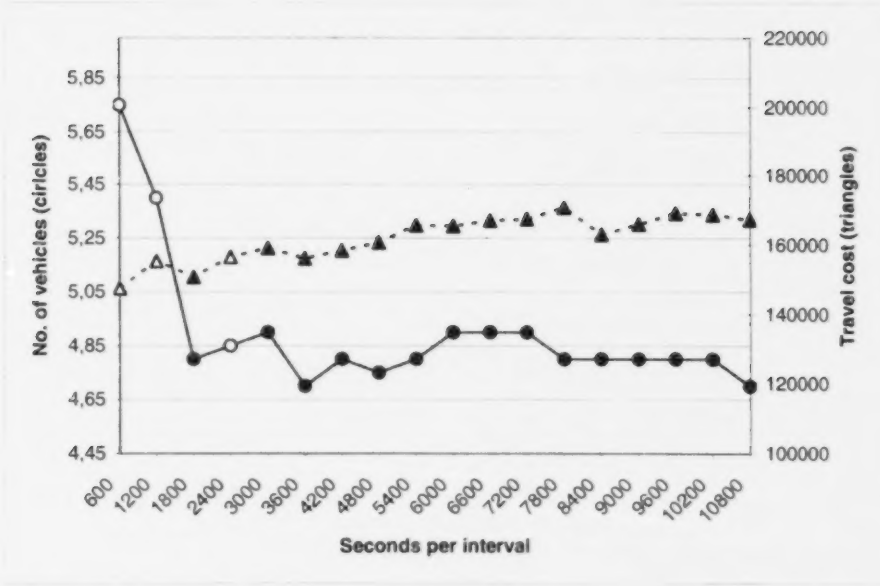


Figure 5: Number of stages vs. solution quality (DSHH)

Although the DSHH seems to have the nice property that increasing the effort (more sample scenarios or more stages) leads to improved solutions, this does not seem to be the case with the pure dynamic heuristic. The effect of spending more time improving the solution by local search is illustrated in Figure 6, where the local search is run between ten and 150 seconds for each interval. It is not clear whether anything is gained by increasing computing time. That is, a better solution to the static subproblems of the algorithm does not necessarily lead to a better overall solution. This is possibly explained by the fact that the local search heuristic produces solutions of a deterministic model where the minimisation of travel cost is based on the known parameters only. As these solutions are improved with respect to travel cost, the flexibility to alter the routes to accommodate future requests may decrease. As noted in Wallace (2000), some ways of solving a problem are better for later periods than others, and deterministic methods will consistently make wrong choices unless flexibility comes for free.

Similarly, Figure 7 reveals that travel costs actually become higher when the number of stages increases. In fact, the best travel cost is found with an interval size close to two hours, whereas vehicle usage seems to be randomly distributed between 4.55 and 4.65 throughout the entire span of time intervals. Again, the probable explanation may lie in flexibility: when it batches more customers together the algorithm is more likely to find short routes than when it considers only a few new customers per time interval. In the latter case one would start servicing new customers immediately, rather than wait for potential nearby customers to appear. Therefore, as long as the customer time windows are not crucial, one may postpone their service and profit from increased flexibility. Similar observations were made by Mitrović-Minić and Laporte (2003) who have studied waiting strategies in the context of the dynamic *Pick-up and Delivery Problem with Time Windows*.

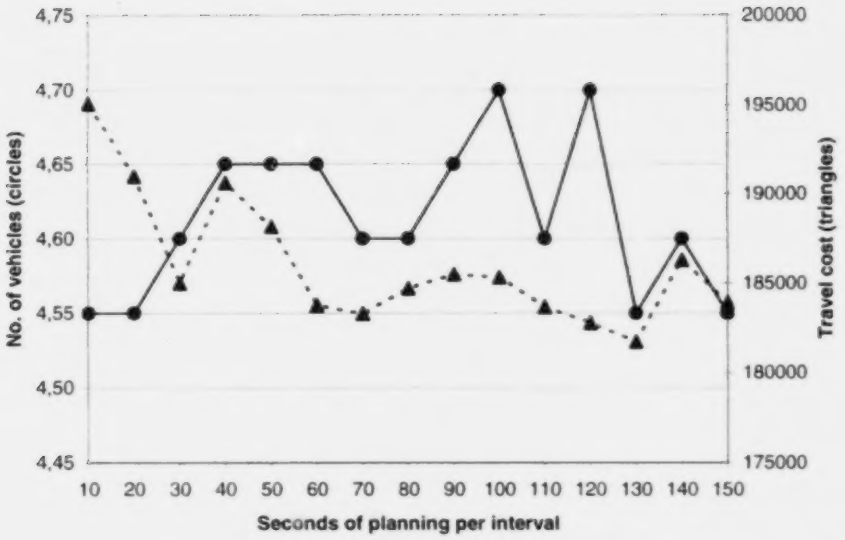


Figure 6: Seconds of planning vs. solution quality (dynamic heuristic)

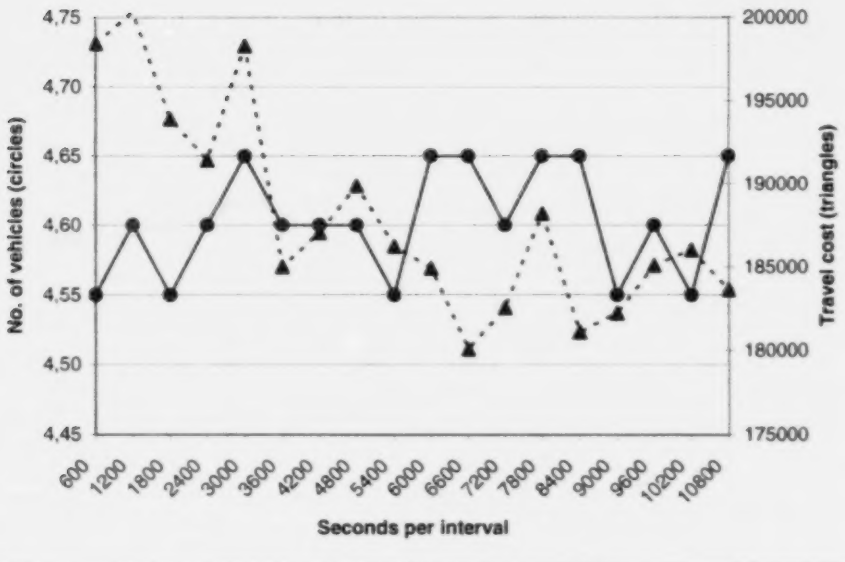


Figure 7: Number of stages vs. solution quality (dynamic heuristic)

In the test runs presented so far, the DSHH were supplied with the same distributions as those used when creating the problem instances, allowing the sample scenarios to reflect real demand very precisely. However, in a real-world setting the real parameter distributions are unknown, and approximations must be used. Figure 8 shows how the DSHH responds when the wrong

number of expected customers is given as input, with values in the 0 to 200 range. There is a clear trend in solution quality as the expected number of customers in the sample scenarios is changed: travel cost steadily decreases when more customers are included in the sample scenarios, even when including far more customers than are actually expected to appear. One may conjecture that the excess customers contribute to decrease travel costs in two ways: by more accurately reflecting the potential customer locations in each sample scenario, and by facilitating the postponement of servicing customers until more information is revealed.

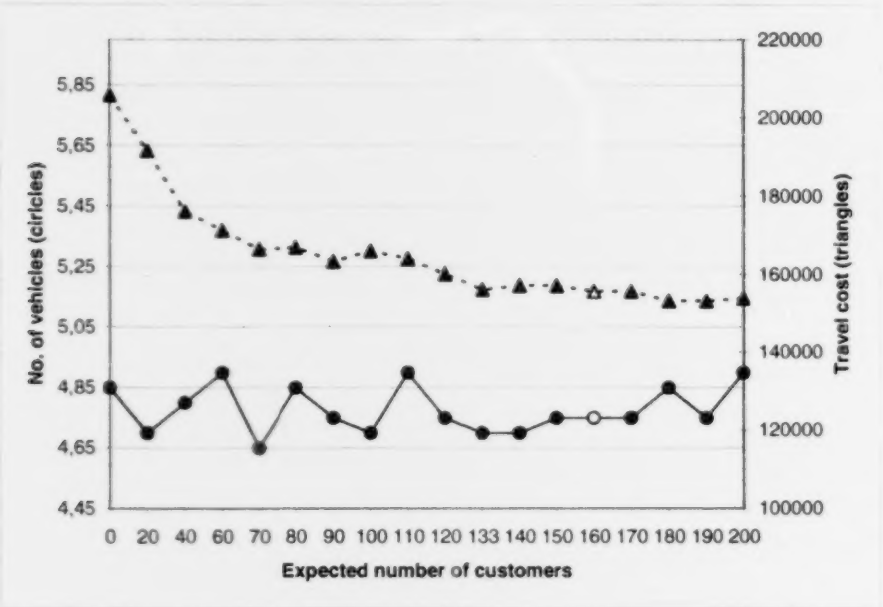


Figure 8: Number of customers in sample scenarios vs. solution quality (DSHH)

Even though the same customer base – those customers already known – is present in each sample scenario, a sufficient number of stochastic customers need to be generated in order to make the sample scenario based algorithm function properly. Figure 8 illustrates that travel cost increases noticeably when the number of additional stochastic customers becomes too low. In a sense, reducing the expected number of customers in the sample scenarios to zero downgrades the method to a pure dynamic approach in which stochastic information is disregarded, and in which the problem sub-solutions are of poor quality.

7 Conclusions and future work

We have considered a class of SVRPs in which customers may appear at unknown locations with random demand at any time during the given time horizon. The problem was formulated as a multi-stage stochastic programming problem, having properties associated with both DVRPs and SVRPs.

A heuristic solution method (DSHH) was developed, based on sampling and on ideas borrowed from progressive hedging; sample scenarios were solved using static methods and common features in the sample scenario solutions were used to generate routes. The approach was

assessed against a myopic dynamic heuristic (MDH), compared to which the DSHH was able to reduce travel distances by more than 15%. It was also noted that, unlike the dynamic approach, the solution method seems to possess the property of producing better solutions when increasing the computational effort. This indicates that incorporating stochastic information in dynamic solution methods can help improve solution quality.

Although some interesting results were obtained, there exist ample opportunities for future work. Perhaps the most relevant avenue is to further examine the occasional use of extra vehicles in solutions produced by the DSHH. The idea here is that although implementing common features from sample scenario solutions is most likely good, it may become relatively poor (e.g., necessitating use of an extra vehicle) in a few other realizations. Thus, rather than always implementing the features discovered in phases 2.2 and 3.3 of the DSHH, one could monitor the significance of each decision by computing sample scenario solutions both when the decision is prohibited as well as when it is enforced. The best alternative obtained by averaging over all sample scenarios, could then be selected. This approach bears some resemblance with stochastic branch and bound (see Klein Haneveld and van der Vlerk, 1999), but where only one path in the branching tree is followed and where the branching criterion is selected as in the current DSHH.

Finally, observe that some deficiencies of the DSHH should be fixed before tackling real-world problems. First, one needs an implementation in which the effect of computation time is not ignored; the planning must start some time before the execution of the routes, so that the solution is ready in time to inform the drivers of the potential changes. Secondly, the cost function used in this study is rarely appropriate in real-world situations where hourly wages, variable expenses per kilometer and the cost of leasing extra vehicles may warrant other cost calculations altogether.

Appendix

The results of Section 6, as obtained by the sample based approach and the pure dynamic method, are quite dissimilar. At the same time both methods produce solutions very different from the solutions to the related static problem, in which all information is supposed to be available from the beginning of the solution process. This section attempts to highlight these differences by stepping through the solution procedures for a particular problem instance.

To illustrate, a small instance was generated in which the expected number of customers was set to 30 and the capacity of the vehicles was reduced correspondingly, with other parameters of the distributions of the random variables being equal to those previously used. The resulting instance was then solved by the three approaches mentioned above, dividing the time horizon into ten equal time intervals (stages) when using the pure dynamic and the sample based method.

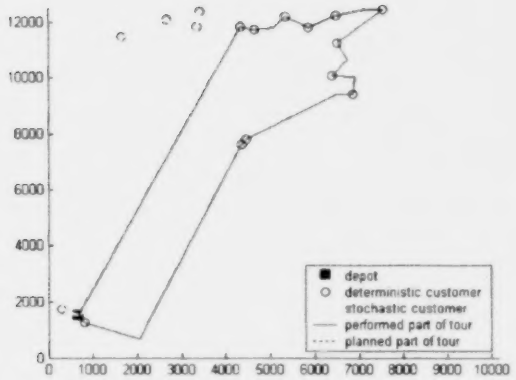
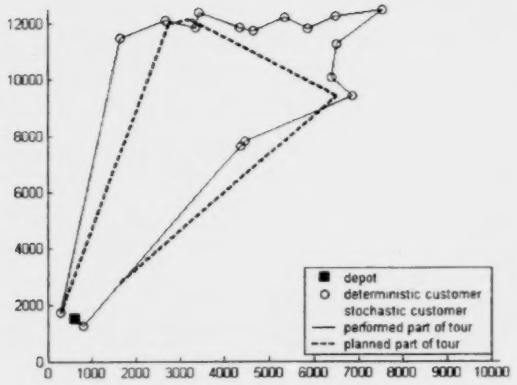
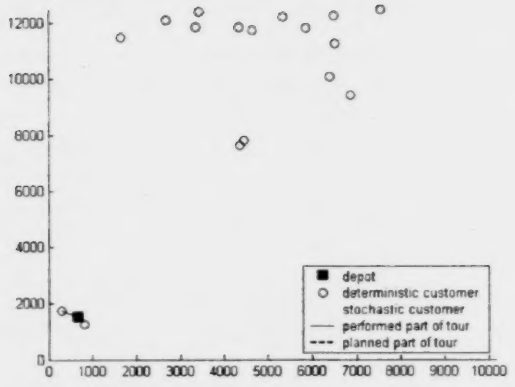


Figure 9: Static solution

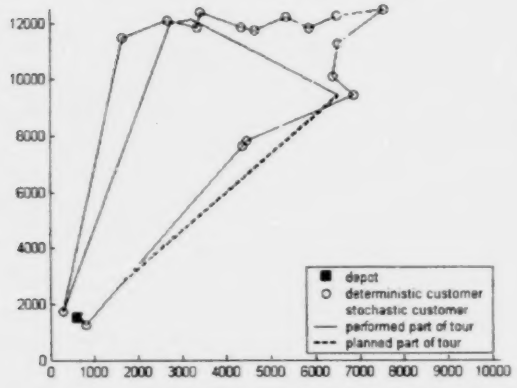
A solution to the corresponding static problem is shown in Figure 9. It uses two vehicles (grey and black lines respectively) and has a length of 55705. The stochastic customers are marked with a grey triangle, whereas the customers known from the start are marked with black circles. The depot is indicated by a black, solid square close to the coordinates (500, 1500). Although this instance is small, it was solved heuristically, and it is therefore unknown whether the solution is optimal. The two routes cross, but this may be caused by some capacity or time constraint, rather than by the sub-optimality of the solution. Solving the same instance using the pure dynamic approach described in Section 6 gives a different impression. In the solution (see Figure 10e), two vehicles are used with a combined travel distance of 94208, i.e., almost 70% more than in the static solution.



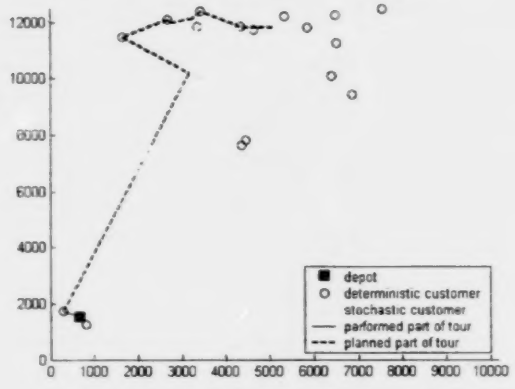
a: Solution after 2 intervals (MDH)



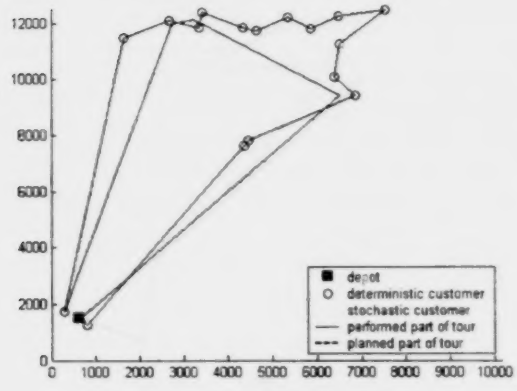
b: Solution after 2 intervals (DSHH)



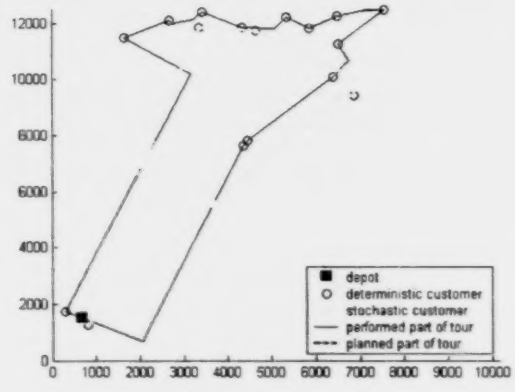
c: Solution after 3 intervals (MDH)



d: Solution after 3 intervals (DSHH)



e: Final solution (MDH)



f: Final solution (DSHH)

Figure 10: Evolution of the solution with MDH and DSHH

The routes depicted in Figure 10e seem worse than what could have been expected from the solution of the static problem. However, the stochastic customers make planning more difficult and the MDH makes poor decisions on several occasions. Figure 10a shows the plan suggested

by the dynamic method after two time intervals have elapsed (corresponding to the third stage of the multi-stage stochastic programming model where still only one vehicle has been used). The parts of the route already performed are marked by solid lines and the parts planned for the remainder of the day are marked by dashed lines. This is the first time that new customers have arrived and, even though the parts of the route constructed so far looks sensible (the solid lines), the new customers force the vehicle back into areas previously visited.

Figure 10c shows the situation for the next stage. The first vehicle has performed most of the work assigned at the previous stage but is now full and a second vehicle must be sent to service another batch of new customers. These customers are again in areas previously visited, and backtracking is unavoidable.

The dynamic stochastic hedging heuristic of Section 5 exhibits a very different behaviour. Instead of planning to consecutively serve the customers without interruptions, the method focuses on which customers to serve immediately in the next time interval. Thus, after two intervals have elapsed, only one customer has been serviced (Figure 10b). The DSHH is able to postpone service to the remaining customers since the downscaled problem instance is easily satisfied with respect to its time window constraints. Note that here only the part of the plans that falls within the next time interval is shown in the illustration (since this is the only part explicitly designed by the heuristic – see Box 1).

The DSHH decides to start the second vehicle at the same stage as the MDH (see Figure 10d), but for a different reason. A second vehicle is started based on the sample scenario solutions, not because the current capacity is exceeded but because future demand would most likely exceed the existing vehicle capacity. In the final plan in Figure 10f two vehicles are used for a total distance of 59437.

The static solution was produced in 30 seconds; the pure dynamic approach required 330 seconds, whereas the DSHH needed less than 12 seconds, in this case with $r = 60$ sample scenarios being generated in each loop of the procedure. To conclude, although this example is scaled down and thus may lead to exaggerated behaviour, the solution methods produce results similar to what was expected based on prior results.

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under grant 00039682. This support is gratefully acknowledged.

References

Ahmed, S. and A. Shapiro. (2002). "The Sample Average Approximation Method for Stochastic Programs with Integer Recourse". Submitted for publication. Technical Report, School of Industrial & Systems Engineering, Georgia Institute of Technology.

Bertsimas, D. J. and G. van Ryzin. (1991). "A Stochastic and Dynamic Vehicle Routing Problem in the Euclidian Plane". *Operations Research* 39, pp. 601-615.

Bertsimas, D. J. and G. van Ryzin. (1993). "Stochastic and Dynamic Vehicle Routing in the Euclidian Plane with Multiple Capacitated Vehicles". *Operations Research* 41, pp. 60-76.

Cordeau, J.-F., G. Desaulniers, J. Desrosiers, M.M. Solomon and F. Soumis. (2002). "The VRP with Time Windows". In: *The Vehicle Routing Problem*, SIAM, Philadelphia. Eds.: P. Toth and D. Vigo, pp. 157-193.

Dror, M., G. Laporte and P. Trudeau. (1989). "Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks". *Transportation Science* 23, pp. 166-176.

Gendreau, M., G. Laporte and R. Séguin. (1995). "An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Customers". *Transportation Science* 29, pp. 143-155.

Gendreau, M., G. Laporte and R. Séguin. (1996a). "Stochastic Vehicle Routing". *European Journal of Operational Research* 88, pp. 3-12.

Gendreau, M., G. Laporte and R. Séguin. (1996b). "A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers". *Operations Research* 44, pp. 469-477.

Gendreau, M. and J.-Y. Potvin. (1998). "Dynamic Vehicle Routing and Dispatching". In: *Fleet management and logistics*, Kluwer. Eds.: T.G. Crainic and G. Laporte, pp. 115-126.

Ichoua, S., M. Gendreau and J.-Y. Potvin. (2001). "Exploiting Knowledge about Future Demands for Real-Time Vehicle Dispatching". Technical report CRT-2001-45, Centre for Research on Transportation, Montreal.

Jagannathan, R. (1985). "Use of Sample Information in Stochastic Recourse and Chance-Constrained Programming Models". *Management Science* 31, pp 96-108.

Kall, P. and S. W. Wallace. (1994). *Stochastic Programming*. Wiley, New York.

Kenyon, A. and D. P. Morton. (2003). "Stochastic Vehicle Routing with Random Travel Times". *Transportation Science* 37, pp. 69-82.

Klein Haneveld, W.K. and M.H. van der Vlerk. (1999). "Stochastic Integer Programming: General models and algorithms". *Annals of Operations Research* 85, pp. 39-57.

Laporte, G., F.V. Louveaux and L. Van hamme. (2002). "An Integer L-Shaped Algorithm for the Capacitated vehicle Routing Problem with Stochastic Demands". *Operations Research* 50, pp. 415-423.

Larsen, A. (2000). "The Dynamic Vehicle Routing Problem". Ph. D. Dissertation, Technical University of Denmark.

Løkketangen, A. and D. Woodruff. (1996). "Progressive Hedging and Tabu Search Applied To Mixed Integer (0,1) Multi-Stage Stochastic Programming". *Journal of Heuristics* 2, pp. 111-128.

Mitrović-Minić, S., R. Krishnamurti and G. Laporte. (2003). "Double-Horizon Based Heuristics for the Dynamic Pickup and Delivery Problem with Time Windows". Forthcoming in *Transportation Research B*.

Mitrović-Minić, S., G. Laporte. (2003) "Waiting Strategies for the Dynamic Pickup and Delivery Problem with Time Windows". Forthcoming in *Transportation Research B*.

Psaraftis, H.N. (1988). "Dynamic Vehicle Routing Problems". In: *Vehicle Routing: Methods and Studies*, Elsevier, North-Holland, Eds.: B. L. Golden and A. A. Assad, pp. 223-248.

Psaraftis, H.N. (1995). "Dynamic Vehicle Routing: Status and Prospects". *Annals of Operations Research* 61, pp. 143-164.

Rockafellar, R. T., and R. J-B. Wets. (1991). "Scenarios and Policy Aggregation in Optimization Under Uncertainty". *Mathematics of Operations Research* 16, pp. 119-147.

Toth, P. and D. Vigo (eds.). (2002). *The Vehicle Routing Problem*. SIAM, Philadelphia.

Verweij, B., S. Ahmed, A. J. Kleywegt, G. L. Nemhauser and A. Shapiro. (2003). "The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study". *Computational Optimization and Applications* 24, pp. 289-333.

Wahl, R., M. Flø, T. Moen, B. Johannesen, T. Engen and T. Haugen. (2002). "Elektronisk datafangst for godstransport i byområder", *SINTEF Report*. (In Norwegian)

Wallace, S. W. (2000). "Decision Making under Uncertainty: Is Sensitivity Analysis of Any Use?" *Operations Research* 48, pp. 20-25.